WEST

Generate Collection Print

L42: Entry 1 of 2 File: USPT May 15, 2001

DOCUMENT-IDENTIFIER: US 6233688 B1
TITLE: Remote access firewall traversal URL

Abstract Paragraph Left (1):
The invention provides a generic naming scheme for remote access and firewall traversal in the form of a uniform resource locator (RAFT URL). The RAFT URL may be provided to any client, regardless of compatibility with the remote access/firewall traversal method, which then launches an operating environment code module. The operating environment code module performs the remote access/firewall traversal method and interacts with the operating environment to obtain data transport mechanisms. These mechanisms permit the client application to transact with private resources beyond the firewall. The remote access/firewall traversal procedure is made transparent to the client application, and thus, a wider array of client applications may be chosen for the data session with the resources beyond the firewall.

Brief Summary Paragraph Right (4):
From within a corporate "intranet" network or a shared private network, the methods and protocols for local area access to computers, devices and data resources within the network is well defined and somewhat uniform, under the control of the administrators of those networks. When users attempt to gain access to those same devices, computers and resources from outside the network, such access is referred to as "remote access". In the past, the most popular physical topology for remote access is a direct dial into a modem bank, which may be at the corporate site or provided for by an ISP (Internet Service Provider). However, this topology impose a heavy administrative burden and monetary cost especially when remote access is attempted through long distance or international toll calls. Thus, there has been a recent trend to provide remote access through an Internet connection. With Internet remote access, the IP (Internet Protocol) connection can be obtained first using any available method, and thus the intranet does not need to maintain a direct physical access point such as a dial-in modem bank. Once a user is "on the Internet" (has achieved an IP connection), a multitude of different protocols and services (limited by the connectivity features of the intranet) can be used on the user's "client" to gain remote access into the intranet. In order to gain remote access, the client must pass the intermediary step, in most cases, of traversing a firewall. The traversal of the firewall can be achieved by using gateway specific software, SSL (Secure Sockets Layer) mechanisms and so on.

Brief Summary Paragraph Right (5):
Specific client software must have support and awareness of specific firewall traversal methods, and thus generic client software cannot be utilized to penetrate the intranet. For example, a client application such as Netscape.TM. may not be able to traverse the firewall since it lacks the means with which to express entry parameters to "support" the private intranet's firewall scheme. Thus, users are often limited to using software that specifically understands and communicates with the intranet. This restricts the choice of client software greatly such that only a limited set of client applications out of all the multitude of programs available can be used when accessing that private intranet.

Brief Summary Paragraph Right (6):
These schemes typically tie the firewall access mechanism to the application, instead of making it transparent by placing it inside the underlying networking support. There is a need for general naming mechanism in order to separate application from firewall traversal mechanisms. Furthermore, the firewall has no standard format to download traversal configuration into the client after authentication.

Brief Summary Paragraph Right (7):
Thus, there is a need for a generic scheme for allowing client applications to be transparent to the remote access and firewall traversal procedure. The scheme should permit any type of remote access/firewall traversal and security method/protocol to be recognized and operated independent of the client application.

Brief Summary Paragraph Right (8):
The invention provides a generic naming scheme for remote access and firewall traversal in the form of a uniform resource locator (RAFT URL). The RAFT URL may be provided to any client application, regardless of compatibility with the remote access/firewall traversal method, which then launches another executable module. The executable module performs the remote access/firewall traversal method and interacts with the operating environment to obtain data transport mechanisms. These mechanisms permit the client application to transact with private resources beyond the firewall. The remote access/firewall traversal procedure is made transparent to the client application, and thus, a wider array of client applications may be chosen for the data session with the resources beyond the firewall.

Drawing Description Paragraph Right (2):
FIG. 1 illustrates the topology of remote access to a private intranet.

Drawing Description Paragraph Right (3):
FIG. 2 illustrates a traditional remote access scheme.

Detailed Description Paragraph Right (2):
FIG. 1 illustrates the topology of remote access to a private intranet.

Detailed Description Paragraph Right (3):
Internet-based remote access may involve many different entities which intervene between a remote access client 110 and hosts such as host 160 and host 170. Remote access client 110 may consist of a computer system or PDA (Personal Digital Assistant) or other information device that has the capability of being connected to a network. Remote access client 110 may also execute software configured to enable the access of services such as FTP (File Transfer Protocol), POP (Post Office Protocol), HTTP (HyperText Transfer Protocol), etc. One way of "getting on the Internet" to commence the remote access session with one or more hosts 160 and 170 is to dial-up to an ISP (Internet Service Provider) whose internal routers and switches provide a TCP/IP (Transport Control protocol/Internet Protocol) connection to remote access client 110. With the "Internet connection thus obtained, the remote access client 110 can reach any computer connected to the "Internet" 130 which is well-known in the art. The ISP dial-up method of Internet access described above is merely exemplary of many ways a connection to the Internet may be obtained.

Detailed Description Paragraph Right (4):
Once on the Internet 130, remote access client 110 is free to access services provided by any computers or networks connected to the Internet 130 such as an HTTP service (i.e., a web site). However, the below description restricts itself to using the Internet to gain remote access into a private network or intranet 150 by means of the Internet. Once remote access into the intranet 150 is established, remote access client 110, services provided for within the intranet 150 may be accessed. What distinguishes the intranet 150 from an ordinary web server or FTP server on a more public network is the security and isolation that can be provided by a gateway or firewall 140. One function of firewall 140 is to prevent unauthorized access into the intranet by users/computers connected to the Internet 130.

Detailed Description Paragraph Right (5):
To control and configure access to the intranet 150 through firewall 140 many various firewall protections or security schemes have been developed such as IP security schemes using SKIP (Simple Key Management for Internet Protocols), or ISAKMP (Internet Security Association and Key Management Protocol), SSL (Secure Sockets layer), etc. Many of these schemes are conflicting and standardization has not been successfully achieved. One reason for the failure of standardization is the nature of remote access--it is intended for a specific often closed set of users. For instance, a company X may desire that only certain key employees have remote access to the intranet of X. In that case, company X will choose whatever method of remote access security is easy to implement or whatever method is decided on as best for the type of information served. Since the choice of remote access security methodology is isolated to the company implementing it, standardization is

difficult. In a more public network such as the Internet standardization is easy to achieve since the many nodes of the network desire compatibility with the other.

Detailed Description Paragraph Right (6):
Lack of standardization limits the remote access client's 110 choice of remote access security methods when using software to access services provided by the intranet.

Detailed Description Paragraph Right (7):
The invention in its various embodiments permits the use of a wider range of client software to access the intranet 150. A naming scheme is provided which when used by client software will cause the client 110 to negotiate remote access. The naming convention is generalized so that any of the conventional security methods can be adequately identified. In one embodiment, the client software parses a RAFT URL (Remote Access Firewall Traversal Uniform Resource Locator) and executes code that extends the ability of the underlying operating system of remote access client 110 to negotiate access according to the RAFT URL. The invention in certain embodiments makes transparent the process of firewall traversal. This transparency will allow the use of application software that is not restricted by or concerned with the remote access security and traversal method.

Detailed Description Paragraph Right (8):
Currently, there is no transparency between the procedure of firewall traversal for remote access and the client application software used to thereafter access the services provided by the intranet beyond the firewall, if the traversal method uses application layer traversal mechanisms. FIG. 2 illustrates this prior art model of application dependency upon the remote access security method. A client application 210 is executed on a system somewhere outside in a topology sense from the firewall 250 and intranet.

Detailed Description Paragraph Right (9):
In traversal methods like httpstunneling, client application 210 must be an application specifically aware and capable of the remote access security and traversal method. The particulars of this method are predefined by the gateway 250 and cannot be modified. As such, if an application such as a web browser does not have built-in support for the remote access method, it will be unable to gain remote access. The process of attaining remote access and/or traversing a firewall is thus, within the purview of the client application in application layer transversal mechanisms.

Detailed Description Paragraph Right (11):
For firewalls in general, RAFT URLs provides a universal language that aids in unifying firewall technology. Currently, some firewalls are based on session layer or application layer mechanisms (application layer gateways or ALGs), whereas other firewalls operate at lower layers, in a much more transparent (to applications) fashion, i.e., IPSEC firewalls, network layer tunneling firewalls. RAFT allows these firewall technologies to be treated in a similar fashion, and to allow their integration and tracking within one single mechanism.

Detailed Description Paragraph Right (13):
Unlike the prior art model of FIG. 2, the client application 310 does not have the responsibility of directly securing remote access. Rather, the client application 310 makes use of a socket factory 320 (Application Program Interface) to access a system resource such as sockets. The socket factory 320 negotiates security and access to gateway/firewall 350. The socket factory is configured to understand the naming convention (RAFT URL) and then initiate steps to obtain remote access, which includes the negotiation of security protocols defined by the RAFT URL. The RAFT URL is input either by user or by preference setting to the client application 310. The socket factory recognizes the RAFT URL (see FIG. 5) and configures itself 330 to communicate with gateway/firewall 350.

Detailed Description Paragraph Right (14):
This socket factory 320 can be made available to any other client application as well, such as a client application 315 or 312. Like client application 310, client applications 312 do not need to be compliant or compatible with the firewall security method. When client applications 312 or 315 obtain sockets via the socket factory, these applications will inherit the same behavior as provided in the sockets to transmit/receive information from gateway/firewall 350. Once the socket factory has successfully gained access beyond the firewall, the sockets derived from

the socket factory 330 provide presentation layer data transport mechanisms to client application 310.

Detailed Description Paragraph Right (15):
Client application 310 (or 312 or 315) would be unaware, except for the obtaining of behavior from the sockets of the firewall traversal. Once client application 310 has obtained the right to transact via the sockets mechanism to gateway/firewall 350, other applications 312 and 315 can use sockets in a similar fashion to communicate beyond the firewall 350. The traversing of the firewall is divorced from the client application 310 and made the responsibility of the socket factory 320. In so doing, traversal of the firewall and its security method is made transparent to client applications 310, 312 and 315. Unlike the prior art of FIG. 2, this permits the client application to be less specific to the firewall.

Detailed Description Paragraph Right (17):
The RAFT URL is a naming scheme generic to the various firewall traversal and remote access security methods. Whatever method the particular firewall designates, the RAFT URL has a structure that can contain identifiers for the underlying implementation (socket factory in the Java case) to handle its negotiation.

Detailed Description Paragraph Right (18):
FIG. 4a shows a system application 400 like one used to configure the socket factory which accepts as input or can store as preference a RAFT URL 410. Unlike an ordinary URL, the RAFT URL does not point to a data object or data creator. Rather, it designates the means to negotiate firewall access which includes specifying security methods either implicitly or explicitly. In one embodiment, the RAFT URL 410 has the following components:

Detailed Description Paragraph Right (20):
Like FIG. 4a, both the designation of "raft:" and a raft-type are provided. Instead of a traversal point, a generic-URL parameter terminating it. Such an embodiment for the RAFT URL may be used in https or secure http protocols, where the "https" URL designates the entry point through the firewall.

Detailed Description Paragraph Right (21):
Described in a listing of desirable implementations of the RAFT URL for specific remote access and firewall traversal schemes. The RAFT URL concept presented in its various embodiments permits any type of scheme to be designated.

Detailed Description Paragraph Right (23):
The RAFT URL for SSL tunneling would take the form "raft:sslt:https://x" where x may designate a host port or server address, directory path and search/cgi (common gateway interface) or file name parameters.

Detailed Description Paragraph Right (24):
2. Mobile IP Firewall Traversal

Detailed Description Paragraph Right (25):
The RAFT URL for firewall traversal by mobile IP (Internet Protocol) would take the form "raft:mip://traversal-point" with an optional terminating string ";type=y", where y refers to either the "SKIP" (Simple Key Management for Internet Protocols) security protocol or IP security protocols.

Detailed Description Paragraph Right (26):
3. Remote Access Using TSP

Detailed Description Paragraph Right (27):
The RAFT URL for remote access using TSP (Tunneling Set-up Protocol) takes the form "raft:tsp://x" where x includes a traversal point and optional ";type=y" terminating parameter. Y may be either "ipsec" or "skip" as defined for Mobile IP access in part 2 above.

Detailed Description Paragraph Right (28):
For instance, consider the following RAFT URL"
"raft:sslt:https://firewall.foo.com/access.html". This RAFT URL indicates that remote access through the firewall named/addressed as "firewall.foo.com" is to be achieved using secure http by processing the page "access.html" which may be an interactive login page where a user enters login information such as a user name and password for further entry beyond the firewall.

Detailed Description Paragraph Right (29):
The RAFT URL is useful due to its versatility; any type of remote access may be designated and specified. A second feature of the invention lies in the processing of the RAFT URL in a manner transparent to the client data application.

Detailed Description Paragraph Right (31):
The first step in creating a transparency between the firewall traversal for remote access and the client application is to obtain the appropriate RAFT URL (step 510). The discovery of the specific RAFT URL to use is not a subject of the essential invention. Obtaining a RAFT URL may be achieved in several ways: (1) obtaining it in person from a system administrator, (2) visiting a special web page where an authenticated user may retrieve the appropriate RAFT URL from the firewall, (3) querying a directory service such as LDAP (Lightweight Directory Access Protocol) or (4) may be preconfigured into the client application or system. The appropriate RAFT URL will designate parameters allowing the client system to get private intranet resources through its data transport mechanisms (IP stack, sockets, etc.).

Detailed Description Paragraph Right (32):
Assuming that the RAFT URL is obtained, it is then provided through some interface to the client application (step 520). This interface may be URL data entry dialog of a browser or be chosen from a menu by the user. The RAFT URL may already be provided to the client application by being set-up in a preference manager for the client application or in an operating system registry intended to service that client application. When so provided, the RAFT URL is passed to the socket factory (in Java) (step 530) that will execute methods needed to perform the firewall traversal procedure.

Detailed Description Paragraph Right (33):
If the specific "raft-type" (type of firewall traversal and/or remote access security, see above) is not provided for in the socket factory, by way of methods, functions, classes and/or code that can be executed, then the required methods, functions, classes, code, etc. must be obtained. In that case, the firewall traversal procedure first gets the needed methods, classes, etc. for the raft-type. These methods, classes, etc. may be obtained from the firewall itself and then loaded as an API (Application Program Interface) or mobile code, such as Java applet, onto the client system. If the required methods, classes, etc., to undergo processing of the "raft-type" is available or made available then the remote access method is performed (step 550). The remote access and/or firewall traversal method, as specified by the "raft-type" parameter of the RAFT URL, need not be known or compatible with the client application(s) ultimately handling the data transactions with resources beyond the firewall. This responsibility and restriction is removed to the socket factory.

Detailed Description Paragraph Right (34):
If the traversal or remote access is successful (checked at step 560), the data transport resources are provided to the client application that extend into the intranet (step 570). The data transport resources may be sockets, a TCP/IP stack or other presentation/transport layer mechanisms which facilitate data transport between the client system and resources such as servers existing in the intranet beyond the firewall. Access is regulated by and monitored by the firewall. The interaction between the API/applet and the obtained data transport resource is dependent upon platform and operating system and will vary from system to system.

Detailed Description Paragraph Right (35):
For example, consider a mobile network computer system based upon Java. Such a computer system typically uses a socket factory to create network data transport resources (called sockets). These sockets allow applications to transact data over a network such as the Internet. Currently, firewall traversal and remote access security operates above this transport layer on the application layer. As a consequence, the choice of client applications restricted when transacting data within an intranet beyond the firewall since compatibility is required. SSL attempts to generalize secure access but does so on the application level and thus, the client application too must be SSL compatible. Further, if SSL is not offered by the firewall as the secure access method, then SSL fails to be a general solution which will be adequate for every firewall traversal and remote access security situation. In the Java based mobile computer system, the invention, in one embodiment, would operate to set the socket factory according to parameters in the RAFT URL. Client applications that use the standard network resources Java application (known as

java.net) can also be divorced from having to include the methods, functions, classes, etc. needed to perform remote access in accordance with the RAFT URL. Rather, a system applet would parse the RAFT URL and set the socket factory by adding methods, functions, classes (if not already present) and then allow the socket factory to handle the firewall traversal and remote access. If a client application wishes directly control its remote access security scheme, it may directly add the required behavior (methods, functions, classes, etc.) to configure its use of the socket factory provided for in the operating system. The actual negotiation of remote access or firewall traversal is made transparent to the application layer, which takes advantage of data transport resources after firewall traversal is successfully complete.

Detailed Description Paragraph Right (36):
The above example refers to a Java-based mobile computing platform. However, the invention, in its various embodiments, is not limited to any platform or operating environment. An application or other network data transport resource can be provided with the means to accept and parse the RAFT URL and then carry out the identified remote access procedure. If the remote access behavior is not available to the operating system, it may be obtained automatically by the application.

Detailed Description Paragraph Right (38):
A RAFT URL mechanism would allow a remote node 610 to traverse into gateway/firewall. Remote node 610 is illustrative of a computer system or information device that desires remote access to a private or other network that lies beyond the firewall. Remote node 610 may be connected to gateway/firewall via a network 600 such as the Internet.

Detailed Description Paragraph Right (39):
Remote node 610 may be composed of a variety of devices, including a memory 614 and a processor 612 coupled directly to each other and through a bus 615. Bus 615 may also attach a network interface card 616 to both memory 614 and processor 612. Network interface card 616 connects the remote 610 to network 600 and allows remote node 610 to transact data to and from network 600 and consequently, to and from other nodes that may be connected to network 600 such as the gateway/firewall. A RAFT URL that identifies the firewall and its security access scheme may be provided to remote node 610 in a variety of ways, including transmission over network 600. Once the appropriate RAFT URL is provided to remote node 610 the RAFT URL is passed to a socket factory generated by some application running in memory 614 and executed by processor 612. The processor 612 is configured (by instructions provided thereto) to parse the RAFT URL and initiate the appropriate client events on remote node 610 for traversal of gateway/firewall 620. Once remote access is granted to remote node 610, it is free to transact data with resources beyond the firewall 620. This data transaction is handled by network interface 616 which may modify data packets with any security-related headers or encapsulation demanded by the RAFT URL's designation.

Detailed Description Paragraph Type 1 (1):
"raft"--indicates the identifying information to follow are handles to configure remote access. Unlike "http:" or "ftp:" of an ordinary URL which identifies a data transfer protocol or service, the "RAFT:" designation can serve to launch or call the remote access mechanism.

Detailed Description Paragraph Type 1 (2):
"raft-type"--designates the particular name given to a specific firewall traversal or remote access method. For instance, the use of layer 3 tunneling with SKIP (Simple Key-Management for Internet Protocols) or tunneling through the firewall using SSL.

Detailed Description Paragraph Type 1 (3):
"traversal point"--indicates the IP address, domain name or other location of the firewall, gateway, or remote access server with which the client system must negotiate access, the traversal point will be known to an authorized user or can be provided through some other means through authentication.

Detailed Description Paragraph Type 1 (4):
"other-info"--indicates a security scheme specific initialization string such as a password, user name or even a secondary security mechanism. The parameter is optional with its format defined wholly by the scheme and firewall's policy.

CLAIMS:

1. A method of remote access comprising:

providing a remote access firewall traversal (RAFT) uniform resource locator (URL) to a client, said RAFT URL indicating a mode of remote access through a firewall; and

configuring said client to negotiate access to a private resource protected by said firewall based on parameters specified by said RAFT URL that allow said client to access said private resource through its data transport mechanisms.

2. A method according to claim 1 wherein the step of recognizing a RAFT URL includes the steps of:

identifying a RAFT service;

identifying a RAFT type, said RAFT type denoting a specific remote access method.

6. A method according to claim 1 wherein the step of configuring includes the steps of:

building sockets from a socket factory in accordance with said RAFT URL; and

passing the behavior of said sockets from said socket factory to application on said client, said application able to traverse said firewall with the appropriate method.

8. A method according to claim 2 further comprising the step of:

performing said specific remote access method, and if said performing is successful, providing data transport resources that extends to said private resource to said client.

9. A computer readable medium comprising:

instructions when executed by a processor cause said processor to provide remote access firewall traversal, said instructions including a remote access firewall traversal (RAFT) uniform resource locator (URL, said RAFT URL indicating a mode of remote access through a firewall.

WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Term	Documents
(33 AND 43).USPT.	1
(L43 AND L33).USPT.	1

Database: US Patents Full-Text Database; US Pre-Grant Publication Full-Text Database; JPO Abstracts Database; EPO Abstracts Database; Derwent World Patents Index; IBM Technical Disclosure Bulletins

Search: L46

Refine Search | Recall Text | Clear

Search History

DATE: Thursday, June 13, 2002 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L46	L43 and l33	1	L46
L45	L43 and l32	1	L45
L44	l31 and l43	2	L44
L43	security same name$1	3867	L43
L42	l31 and (name$1 or address$2)	2	L42
L41	L40 and l31	0	L41
L40	name near3 separate$1	856	L40
L39	name near3 sepate$1	0	L39
L38	name near3 sepater$1	0	L38
L37	L36 and l31	0	L37
L36	name space	653	L36
L35	L33 and l26	1	L35

L34	L32 and l26	1	L34
L33	6205551.pn.	1	L33
L32	6233688.pn.	1	L32
L31	L30 and l12.ab.	2	L31
L30	L29 and l26	75	L30
L29	l24 and l22 and l12	981	L29
L28	object$1	1882799	L28
L27	L26 and l25	0	L27
L26	entry point$1	11285	L26
L25	L24 and l23	9	L25
L24	securit$	65076	L24
L23	l21 and l22	17	L23
L22	access$	581583	L22
L21	l12 and footprint$1	20	L21
L20	Susser.in.	4	L20
L19	5745910.pn.	1	L19
L18	L17 and footprint	11	L18
L17	L16 and l12	617	L17
L16	context$1	134701	L16
L15	L14 and l12	0	L15
L14	context barrier	14	L14
L13	L12 and l11	0	L13
L12	firewall	2623	L12
L11	6092147.pn.	1	L11
L10	5356172.pn.	1	L10
L9	5113959.pn.	1	L9
L8	4962255.pn.	1	L8
L7	4954381.pn.	1	L7
L6	4830903.pn.	1	L6
L5	4668354.pn.	1	L5
L4	4593137.pn.	1	L4
L3	4584722.pn.	1	L3
L2	4314931.pn.	1	L2
L1	6092147.pn.	1	L1

END OF SEARCH HISTORY

WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Term	Documents
(33 AND 43).USPT.	1
(L43 AND L33).USPT.	1

Database: US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L46

Refine Search

Recall Text | Clear

Search History

DATE: Thursday, June 13, 2002 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L46	L43 and l33	1	L46
L45	L43 and l32	1	L45
L44	l31 and l43	2	L44
L43	security same name$1	3867	L43
L42	l31 and (name$1 or address$2)	2	L42
L41	L40 and l31	0	L41
L40	name near3 separate$1	856	L40
L39	name near3 sepate$1	0	L39
L38	name near3 sepater$1	0	L38
L37	L36 and l31	0	L37
L36	name space	653	L36
L35	L33 and l26	1	L35

L34	L32 and l26	1	L34
L33	6205551.pn.	1	L33
L32	6233688.pn.	1	L32
L31	L30 and l12.ab.	2	L31
L30	L29 and l26	75	L30
L29	l24 and l22 and l12	981	L29
L28	object$1	1882799	L28
L27	L26 and l25	0	L27
L26	entry point$1	11285	L26
L25	L24 and l23	9	L25
L24	securit$	65076	L24
L23	l21 and l22	17	L23
L22	access$	581583	L22
L21	l12 and footprint$1	20	L21
L20	Susser.in.	4	L20
L19	5745910.pn.	1	L19
L18	L17 and footprint	11	L18
L17	L16 and l12	617	L17
L16	context$1	134701	L16
L15	L14 and l12	0	L15
L14	context barrier	14	L14
L13	L12 and l11	0	L13
L12	firewall	2623	L12
L11	6092147.pn.	1	L11
L10	5356172.pn.	1	L10
L9	5113959.pn.	1	L9
L8	4962255.pn.	1	L8
L7	4954381.pn.	1	L7
L6	4830903.pn.	1	L6
L5	4668354.pn.	1	L5
L4	4593137.pn.	1	L4
L3	4584722.pn.	1	L3
L2	4314931.pn.	1	L2
L1	6092147.pn.	1	L1

END OF SEARCH HISTORY

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 1 of 1 returned.

☐ 1. Document ID: US 6233688 B1

L45: Entry 1 of 1 File: USPT May 15, 2001

US-PAT-NO: 6233688
DOCUMENT-IDENTIFIER: US 6233688 B1

TITLE: Remote access firewall traversal URL

DATE-ISSUED: May 15, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Montenegro; Gabriel	Fremont	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Sun Microsystems, Inc.	Mountain View	CA			02

APPL-NO: 9/ 109260
DATE FILED: June 30, 1998

INT-CL: [7] G06 F 11/00

US-CL-ISSUED: 713/201
US-CL-CURRENT: 713/201

FIELD-OF-SEARCH: 713/201, 713/200, 713/202, 707/3, 707/4, 707/100, 707/102, 709/232, 709/237, 380/25, 380/29

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5818019	June 1999	Valencia	395/20057
5822539	October 1998	Van Hoff	395/20066
5944823	August 1999	Jade et al.	713/201
5950195	September 1999	Stockwell et al.	707/4
5987611	November 1999	Freund	713/201
5999979	December 1999	Vellanki et al.	709/232
6061797	May 2000	Jade et al.	713/201
6073176	June 2000	Baindur et al.	709/227
6088796	July 2000	Cianfrocca et al.	713/152

ART-UNIT: 214
PRIMARY-EXAMINER: Iqbal; Nadeem
ATTY-AGENT-FIRM: Blakely Sokoloff Taylor & Zafman

ABSTRACT:

The invention provides a generic naming scheme for remote access and firewall traversal in the form of a uniform resource locator (RAFT URL). The RAFT URL may be provided to any client, regardless of compatibility with the remote access/firewall traversal method, which then launches an operating environment code module. The operating environment code module performs the remote access/firewall traversal method and interacts with the operating environment to obtain data transport mechanisms. These mechanisms permit the client application to transact with private resources beyond the firewall. The remote access/firewall traversal procedure is made transparent to the client application, and thus, a wider array of client applications may be chosen for the data session with the resources beyond the firewall.

9 Claims, 7 Drawing figures

Generate Collection | Print

Term	Documents
(32 AND 43).USPT.	1
(L43 AND L32).USPT.	1

Display Format: FRO | Change Format

Previous Page Next Page

PALM INTRANET

Day : Thursday
Date: 6/13/2002
Time: 13:49:05

Application Number Information

Application Number: **09/235155**
Assignments
Filing Date: **01/22/1999**
Effective Date: **01/22/1999**
Application Received: **01/22/1999**
Patent Number:
Issue Date: **00/00/0000**
Date of Abandonment: **00/00/0000**
Attorney Docket Number: **50253-218**
Status: **41 /NON FINAL ACTION MAILED**
Confirmation Number: **5107**

Examiner Number: **76212 / OPIE, GEORGE**
Group Art Unit: **2151**
Class/Subclass: **709/100.000**
Lost Case: **NO**
Interference Number:
Unmatched Petition: **NO**
L&R Code: Secrecy Code:**1**
Third Level Review: **NO**

Waiting for Response Desc.
CPA Processing
Mail Non Final

Secrecy Order: **NO**
Status Date: **03/11/2002**

Title of Invention: **TECHNIQUES FOR PERMITTING ACCESS ACROSS A CONTEXT BARRIER ON A SMALL FOOTPRINT DEVICE USING RUN TIME ENVIRONMENT PRIVILEGES**

Bar Code	Location	Location Date	Chrg to Loc	Charge to Name	Emp. ID	Infra Loc
09235155	21D3 2100 TEAM 3, PK-2 4W30	06/06/2002		No Charge to Name	DFORTE	PK2/04/W 30

Appln Info | Contents | Petition Info | Atty/Agent Info | Continuity Data | Foreign Data | Inve

Search Another: Application# [ ] Search **or Patent#** [ ] Search

PCT / [ ] / [ ] Search **or PG PUBS #** [ ] Search

Attorney Docket # [ ] Search

Bar Code # [ ] Search

(To Go BACK Use BACK Button on Your BROWSER Tool Bar)

Back to PALM | ASSIGNMENT | OASIS | Home page

PALM INTRANET

Day : Thursday
Date: 6/13/2002
Time: 13:10:39

Inventor Name Search Result

Your Search was:

Last Name = LEVY
First Name = MOSHE

Application#	Patent#	Status	Date Filed	Title	Inventor Name
06157904	4314931	150	06/09/1980	TONER PIGMENT TREATMENT PROCESS FOR REDUCING THE RESIDUAL STYRENE MONOMER CONCENTRATION TO LESS THAN 0 .5 PER CENT BY WEIGHT	LEVY , MOSHE
06493327	4584722	150	05/10/1983	PROSTHETIC TENDON	LEVY , MOSHE
06592795	4593137	150	03/23/1984	PARA-SELECTIVE AND BETA-SELECTIVE CRYSTALLIZED GLASS ZEOLITE ALKYLATION CATALYST	LEVY , MOSHE
06783011	Not Issued	166	10/02/1985	CATALYSTS AND PROCESS FOR THE PRODUCTION OF HYDROCARBONS AND SUBSTITUTION OF HYDROCARBONS	LEVY , MOSHE
06817701	Not Issued	166	01/10/1986	PARA-SELECTIVE AND BETA-SELECTIVE CRYSTALLIZED GLASS ZEOLITE ALKYLATION CATALYST	LEVY , MOSHE
06901770	4668354	150	08/29/1986	CATALYTIC DEPOSITION OF METALS IN SOLID MATRICES	LEVY , MOSHE
06947929	Not Issued	168	12/30/1986	NOVEL METHOD FOR THE PREPARATION OF POROUS SUBSTRATES, WITH A WELL DEFINED MORPHOLOGY, TO BE EMPLOYED IN MEMBRANE SEPARATIONS, MICROENCAPSULATION AND CONTROLLED RELEASE APPLICATIONS	LEVY , MOSHE
07019218	4830903	150	02/26/1987	CATALYTIC DEPOSITION OF METALS IN SOLID MATRICES	LEVY , MOSHE
07087991	Not Issued	161	08/17/1987	PARA-BETA SELECTIVE AND BETA-SELECTIVE CRYSTALLIZED GLASS	LEVY , MOSHE

07093094	Not Issued	166	08/31/1987	ISOMERIZATION OF XYLENE AND CATALYST FOR SUCH PROCESS	LEVY , MOSHE
07171497	4954381	150	03/21/1988	PREPARATION OF POROUS SUBSTRATES HAVING WELL DEFINED MORPHOLOGY	LEVY , MOSHE
07319865	4962255	150	03/06/1989	CATALYSTS AND PROCESS FOR THE PRODUCTION OF HYDROCARBONS AND SUBSTITUTION OF HYDROCARBONS	LEVY , MOSHE
07471074	Not Issued	161	01/29/1990	ISOMERIZATION OF XYLENE AND CATALYST FOR SUCH PROCESS	LEVY , MOSHE
07575222	5113959	150	08/30/1990	ELECTRIC DRIVE ATTACHMENT FOR A WHEELCHAIR	LEVY , MOSHE
07918305	Not Issued	166	07/21/1992	WHEELCHAIRS	LEVY , MOSHE
08141908	5356172	150	10/21/1993	GLIDING SEAT ASSEMBLY FOR A PROPELLED WHEEL CHAIR	LEVY , MOSHE
08419084	Not Issued	161	07/10/1995	DRY MIRROR CONCEPT	LEVY , MOSHE
08839621	6092147	150	04/15/1997	VIRTUAL MACHINE WITH SECURELY DISTRIBUTED BYTECODE VERIFICATION	LEVY , MOSHE
09430524	Not Issued	030	10/29/1999	UNIVERSAL SMART CARD ACCESS SYSTEM	LEVY, MOSHE
09547225	Not Issued	120	04/11/2000	VIRTUAL MACHINE WITH SECURELY DISTRIBUTED BYTECODE VERIFICATION	LEVY, MOSHE
10014823	Not Issued	030	10/29/2001	ENHANCED QUALITY OF IDENTIFICATION IN A DATA COMMUNICATIONS NETWORK	LEVY, MOSHE
10014893	Not Issued	030	10/29/2001	USER ACCESS CONTROL TO DISTRIBUTED RESOURCES ON A DATA COMMUNICATIONS NETWORK	LEVY, MOSHE
10014934	Not Issued	020	10/29/2001	PORTABILITY AND PRIVACY WITH DATA COMMUNICATIONS NETWORK BROWSING	LEVY, MOSHE
10033373	Not Issued	020	10/29/2001	IDENTIFICATION AND PRIVACY IN THE WORLD WIDE WEB	LEVY, MOSHE
10040270	Not Issued	019	10/29/2001	IDENTIFICATION AND PRIVACY IN THE WORLD WIDE WEB	LEVY, MOSHE

Search and Display More Records.

Search Another: Inventor

Last Name	First Name
Levy	Moshe

Search

(To go back use Back button on your browser toolbar.)

Back to PALM | ASSIGNMENT | OASIS | Home page

PALM INTRANET

Day : Thursday
Date: 6/13/2002
Time: 13:10:58

Inventor Name Search Result

Your Search was:

Last Name = LEVY
First Name = MOSHE

Application#	Patent#	Status	Date Filed	Title	Inventor Name
10040293	Not Issued	020	10/29/2001	PRIVACY AND IDENTIFICATION IN A DATA	LEVY, MOSHE
60255569	Not Issued	020	12/14/2000	CROSS NETWORKS CROSS INFRASTRUCTURES (CNCI)	LEVY, MOSHE

Inventor Search Completed: No Records to Display.

Search Another: Inventor

Last Name Levy **First Name** Moshe

Search

(To go back use Back button on your browser toolbar.)

Back to PALM | ASSIGNMENT | OASIS | Home page

WEST

End of Result Set

Generate Collection | Print

L42: Entry 2 of 2 File: USPT Mar 20, 2001

DOCUMENT-IDENTIFIER: US 6205551 B1
TITLE: Computer security using virus probing

Abstract Paragraph Left (1):
A technique for determining whether particular clients within a computer network are universally configured in accordance with the desired network security features of the computer network. A probe is randomly inserted within incoming files, e.g., at a firewall in the computer network. The probe is configured as a function of a particular execution task, e.g. a known virus, such that in a properly configured client the probe will not execute and the firewall does not detect a security breach. However, if the client is misconfigured, i.e., not in compliance with the standard network security features, the probe will execute and trigger an alarm in the firewall indicating that the client is vulnerable to a security breach. Advantageously, a network security administrator can take appropriate action to correct those clients which are misconfigured.

Brief Summary Paragraph Right (1):
The present invention relates to network security and, more particularly, to a technique for the verification of security measures employed in computer networks.

Brief Summary Paragraph Right (2):
Advances in communications technology and the availability of powerful desktop computer hardware has increased the use of computers to access a variety of publicly available computer networks. Today, a tremendous amount of information is exchanged between individual users located around the world via public computer networks, e.g., the Internet. One class of users includes private individuals and professional users interconnected via a private network, e.g., a corporate intranet. The exchange of information between private and public computer networks has presented a variety of critical security issues for the protection of information on the private computer networks and the overall functionality of the private computer network itself.

Brief Summary Paragraph Right (3):
Computer network security, at a minimum, is directed to ensuring the reliable operation of computing and networking resources, and protecting information within the network from unauthorized disclosure or access. Various security threats exist which pose increasingly difficult challenges to such network security. In particular, some of the most sophisticated types of security threats are posed by programs which exploit certain vulnerabilities within network computing systems. To name a few, these program-related security threats include well-known logic bombs, trapdoors, trojan horses, viruses and worms, as described, e.g., by W. Stallings, Network and Internetwork Security Principles and Practice, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1995. Such well-known software program threats either work independently (e.g., worms) to achieve their desired security breach, or require the invocation of a host program to be invoked to perform the desired disruptive actions (e.g., trapdoors, logic bombs, trojan horses or viruses.) Indeed, there are numerous well publicized accounts of such programs being used to improperly breach the security of private computer networks and cause severe damage (see, e.g., J. Hruska, Computer Viruses and Anti-Virus Warfare, Second edition, Ellis Horwood Limited, New York, 1992.) Such damage has included the destruction of electronic files, alteration of databases, or the disabling of the computer network itself or computer hardware connected to the affected network.

Brief Summary Paragraph Right (4):

Network administrators responsible for the operation of private computer networks employ a variety of security measures to protect the network from external security breaches such as the introduction of computer viruses. One technique uses so-called firewalls. This security scheme essentially places a separate computer system, i.e., the firewall, between the private network and the public network, e.g., the Internet. These the firewalls are software-based gateways that are typically installed to protect computers on a local area network ("LAN") from attacks by outsiders, i.e., unauthorized users. The firewall maintains control over communications from and to the private network. Essentially, the firewall imposes certain security measures on all users employing the private network. For example, firewalls may block access to new Internet services or sites on the World Wide Web ("WWW") because the security consequences are unknown or not accounted for by the present firewall configuration. One potential installation configuration of a firewall is that WWW clients can no longer directly contact WWW servers. Typically, this proves too restrictive, and network administrators employ so-called "proxy servers". Proxy servers are designed with certain features which provide for the forwarding of requests from WWW clients through the firewall thereby providing communication flow to and from servers on the Internet.

Brief Summary Paragraph Right (5):
Recently, firewall vendors have included so-called "virus filtering" features to address critical security issues associated with virus infection. More particularly, this virus filtering at the firewall is conceptually similar to well-known virus scanning typically employed on client machines, e.g., personal computers, which reside within a LAN in a conventional client/server arrangement. In such client-based virus detection, virus scanning is accomplished using a program which searches through, e.g., the operating system, executable files, system files, boot records, and memory, of the client looking for the presence of undesirable software entities. Computer viruses are detected by the virus scanner by using previously defined "virus signatures" associated with each virus. The virus signature is typically a fixed-length signature pattern, e.g., a 16 to 24 byte pattern, extracted from the known virus by the vendor of the virus scanning software. The virus scanning software contains a list of signatures for known computer viruses and scans the various files in a particular client looking for a match to a particular virus signature. If a match is found, this entity of the client is "infected" and the user is notified accordingly.

Brief Summary Paragraph Right (6):
The incorporation of virus filtering within commercially available firewalls provides for virus detection by scanning files transmitted through the firewall. While this provides the firewall with additional network security capabilities, implementing the virus filter at the firewall presents certain operational difficulties which include: (1) a substantial amount of processing must be accomplished at the firewall which degrades network performance through the introduction of latency which affects applications executing in the network; and (2) the firewall itself contains less operational and data intelligence with regard to individual clients in the network which leads to a less precise scan of the incoming data by the firewall as could be accomplished by a client-based virus scanner.

Brief Summary Paragraph Right (7):
Therefore, given the potential drawbacks in firewall-based virus filtering, most network security administrators opt for providing virus screening in the client machines across the network rather than in the firewall itself. Currently, a number of popular commercial computer virus scanners are used for such client-based scanning. Typically, network security administrators will select a particular commercially available virus scanning program and install the program across all the clients of the network. Of course, the effectiveness of the virus scanning software is as function of the uniformity of installation and periodically updating the virus signature listing used by the software to included newly identified viruses. As will be appreciated, for very large client/server networks the task of ensuring that the virus detection software is universally installed and updated on all clients is significant and not always achievable. A client-by-client inspection is labor intensive and cannot be undertaken on a frequent enough basis to ensure conformity. Therefore, individual users are typically responsible for updating their virus scanning software by, e.g., downloading the most current virus signature listing from a central source. Of course, the lack of diligence and infrequency of such updates by individual users can lead to potential secure breaches within the network.

Brief Summary Paragraph Right (8):
A need exists therefore for ensuring that network security features are universally configured throughout a computer network.

Brief Summary Paragraph Right (9):
The present invention provides a technique for determining whether particular clients within a computer network are universally configured in accordance with the desired security features of the computer network. In accordance with the invention, a probe is randomly inserted within incoming files in the computer network. Illustratively, the insertion of probes occurs in a firewall which separates the computer network from other networks. The probe, in accordance with an embodiment of the invention, is configured as a function of a particular execution task, e.g. a known virus, such that in a properly configured client the probe will not execute and the firewall does not detect a security breach. However, if the client is misconfigured, i.e., is not in compliance with the standard network security measures, the probe will execute and trigger a security alert in the firewall indicating that the client is vulnerable to a security breach. Advantageously, a network security administrator can take appropriate action to correct those clients which are misconfigured.

Brief Summary Paragraph Right (10):
In preferred embodiments of the invention, the probe is configured as a virus probe in the form of a trojan horse which, if executed, on a client will launch a signal back to the firewall indicating that the client is misconfigured. In further embodiments of the invention, the signal back to the firewall is a User Datagram Protocol ("UDP") packet. In accordance, with a further embodiment of the invention, the virus probe is inserted upon a first Internet access from a particular IP address or browser type, and thereafter virus probes are inserted at random intervals.

Drawing Description Paragraph Right (2):
FIG. 2 is a flowchart of operations illustratively performed by the firewall of FIG. 1 in implementing the present invention, and

Detailed Description Paragraph Right (1):
The present invention provides a technique for determining whether particular clients within a computer network are universally configured in accordance with the desired security features of the computer network. In accordance with the invention, a probe is randomly inserted within incoming files, illustratively, at a firewall in the computer network. The probe, in accordance with an embodiment of the invention, is configured as a function of a particular execution task, e.g. a known virus, such that in a properly configured client the probe will not execute and the firewall does not detect a security breach. However, if the client is misconfigured, i.e., is not in compliance with the standard network security measures, the probe will execute and trigger an alarm in the firewall indicating that the client is vulnerable to a security breach. Advantageously, a network security administrator can take appropriate action to correct those clients which are misconfigured.

Detailed Description Paragraph Right (2):
FIG. 1 shows an exemplary system embodying the principles of the invention. As shown in FIG. 1, the system includes public network 100, e.g., the Internet, and network resources 105, 110, 115, 120 and 125. Illustratively, network resources 105 through 125 can be linked together using files written in the well-known Hypertext Mark-up Language ("HTML") thereby representing the well-known WWW. The WWW and HTML are described in more detail, e.g., by B. White, HTML and the Art of Authoring for the World Wide Web, Kluwer Academic Publishers, Norwell, Mass; 1996. Illustratively, private network 130 is a network located within a particular user site, e.g., a corporation's headquarters building, having user terminals 165-1, 165-2, 165-3 and 165-4 linked together via LAN 170. As will be appreciated, user terminals 165-1 through 165-4 can be stand-alone personal computers or network terminals. For simplicity of explanation herein, only one such LAN configuration is shown in FIG. 1, however, as will be appreciated private network 130 may include several such LAN configurations similar in nature to LAN 170. A particular user of any one of user terminals 165-1 through 165-4 may cause a client program executing on, e.g., user terminal 165-3, to request certain resources which are available on the WWW, e.g., network resources 105-125. As mentioned previously, such requests to the WWW via the Internet from private network 130 pose certain security risks to both private network 130 and user terminals 165-1 through 165-4. Thus, as shown in FIG. 1, private network 130 includes firewall 180 and proxy server 135 which are configured

to delivery certain security features, in accordance with the invention, to protect private network 130 and its various computing resources.

Detailed Description Paragraph Right (3):
As discussed previously, network administrators responsible for the operation of private computer networks, e.g., private network 130, employ a variety of security measures to protect the network from external security breaches such as the introduction of computer viruses. One technique places a separate computer system, i.e., the firewall, between the private network and the public network, e.g., the Internet. The firewall monitors and maintains control over communications from and to the private network. More particularly, where a private network employs a firewall, the firewall first determines if the requested connection between a user terminal in the private network and the public network is authorized. The firewall serves as an intermediary between the user terminal in the private network and the public network and, if the connection is authorized, facilitates the requisite connection between the two networks. Alternatively, if the connection is unauthorized, the firewall prevents any connection between the networks from occurring.

Detailed Description Paragraph Right (4):
In accordance with the illustrative embodiment of the invention shown in FIG. 1, proxy server 135 includes processor 140, web proxy 145, file transport protocol ("FTP") proxy 150 and mail proxy 160. As will be appreciated, these illustrative proxies enable the proxy server, working in conjunction with the firewall, to provide security features for WWW/Internet access, file transfers and electronic mail, respectively. For example, web proxy 145 is used when a user desires to access particular "web pages" on the WWW from private network 130. Illustratively, a user employing user terminal 165-3 may access certain web pages on the WWW using web browser 166. Web browsers are well-known software application programs (e.g., Netscape.RTM. v. 5.0, available from Netscape Communications) which enable a user to traverse the WWW and access the vast amount of information available throughout the WWW. Thus, web browser 166 receives an input request from the user of user terminal 165-3 and attempts to locate the information on the WWW by establishing a connection with the appropriate resource, e.g., network resource 105, on the WWW through public network 100. The connection between user terminal 165-3 and network resource 105 is established using proxy server 135, web proxy 145 and firewall 180. More particularly, web proxy 145, acting on behalf of web browser 166, will attempt to establish a conventional Transfer Control Protocol/Internet Protocol ("TCP/IP") connection between user terminal 165-3 and network resource 105. As is well-known, TCP/IP is the protocol which is used in describing the way in which information is transferred across the Internet. Essentially, TCP/IP separates information into individual packets and routes these packets between the sending computer, e.g., server, and the receiving computer, e.g., client. TCP/IP and Internet communications are discussed in more detail, e.g., by D. Comer., Internetworking with TCP/IP, Third edition, Prentice-Hall, Englewood Cliffs, N.J., 1995. In the present embodiment, the TCP/IP connection between user terminal 165-3 and network resource 105 is made across communication channels 190 and 195, respectively, which establish connection between public network 100, private network 130 and, ultimately, user terminal 165-3.

Detailed Description Paragraph Right (5):
As seen from FIG. 1, all communications traffic between public network 100 and private network 130 necessarily passes through firewall 180. In recognition of this communications traffic attribute, I have realized that the firewall 180 provides a preferred location for implementing the security advantages of my invention. Illustratively, in accordance with the preferred embodiment of the invention, firewall 180 illustratively includes processor 181, database 182, and virus prober 185 which randomly inserts probes within incoming files from, e.g., public network 100, to, e.g., private network 130. In accordance with the invention, the probes inserted by the virus prober 185 are individual programs which will trigger particular actions upon execution. In accordance with an embodiment of the invention, the probe is a virus probe configured as a trojan horse which, if executed, on a client will launch a signal back to the firewall indicating that the client is misconfigured. Typically, from a computer virus perspective, a trojan horse is a secret, undocumented entry point placed into a useful application program by an unauthorized user, e.g., computer hacker. In the normal course of execution of the useful application program by a user the trojan horse is also executed thereby launching the undesired actions. Trojan horses are described in more detail in Stallings, supra. at pp. 238-241. For example, a trojan horse can be created to gain

access to the files of another user on a shared computer system, wherein the unauthorized user creates a trojan horse program that, when executed, changes the authorized user's file permissions so that their files become readable by any user. This embodiment of the invention utilizes particular features of the trojan horse for delivery of various security advantages to computer networks as discussed in more detail below.

Detailed Description Paragraph Right (6):
In accordance with the invention, the virus probe inserted by virus prober 185 at firewall 180 is benign in that the probe is designed to provide a signal back to firewall 180 if executed, rather than perform some destructive action as in the conventional trojan horse sense as described previously. The security features of the invention are preferably implemented and realized at the firewall, e.g., firewall 180, because in networks where firewalls are employed all communications traffic must pass through the firewall. Thus, the firewall is an ideal location for inserting probes in accordance with the invention. However, will be appreciated, the principles of the invention are also realized in other network environments and configurations. For example, in accordance with a further embodiment of the invention, the insertion of probes can be accomplished using a particular proxy server within a network that is known to have a high rate of common access and is trusted. For example, a trusted server within a private network which mainly provides an online telephone directory is also an excellent candidate for implementing the principles of the invention due to the fact that this server will be utilized by a high number of user within the private network. Thus, the security features delivered by the present invention are realized in a variety of network, hardware and software configurations including, but not limited to, the system configuration of FIG. 1.

Detailed Description Paragraph Right (7):
The operations of delivering network security through the insertion, monitoring and execution of probes in accordance with the invention is shown in the illustrative operations of FIG. 2. In accordance with the preferred embodiment of the invention, as described above, the operations of FIG. 2 are initiated within firewall 180. More particularly, in accordance with the invention, the communications traffic stream, e.g., in and out of private network 130, is continually monitored (block 200.) During the course of monitoring the communications traffic stream transmitted across the network, probes are randomly inserted into incoming files (block 205) destined for private network 130. The structural aspects of the probe of the invention are described below in more detail with regard to FIG. 3. In accordance with the invention, the probes are designed, if executed on a client, to trigger a signal indicative of a security alert.

Detailed Description Paragraph Right (8):
Illustratively, the signal can be a request for a network resource. Since all such requests must be made through the firewall, this ensures that when a probe configured in accordance with the invention triggers such a request, the request can effectively be utilized as the signal to the firewall. That is, such signals triggered by the probe will be immediately recognizable by the firewall. In further embodiments of the invention, the signal can be in the form of a conventional User Datagram Protocol ("UDP") packet. As will be appreciated, UDP is a transport protocol which runs on top of the conventional TCP/IP protocol and provides a low overhead mechanism for two applications to quickly exchange small amounts of data. UDP requires less overhead than typical TCP/IP packet exchanges because UDP is a less secure protocol than TCP/IP. That is, UDP is transaction oriented, and packets may be duplicated, lost or received in a different order than as originally sent. In contrast, TCP/IP is more reliable because the protocol goes to significant lengths (e.g., generating checksums, acknowledging the receipt of packets, retransmitting lost packets) to insure that data arrives at its destination intact. Since UDP has no such overhead it is considerably faster than TCP/IP and is ideal for applications, as in various embodiments of the invention, that transmit short bursts of data, need faster network throughput, or do not require verification of delivery at the destination. As will be appreciated, other types of signal configurations, in addition to those described above, which will be equally effective in delivering the various aspects of the invention.

Detailed Description Paragraph Right (9):
Thus, when firewall 180 receives the security alert indication, e.g., UDP packet, that a particular probe has executed (block 210), the firewall will identify the probe and client (block 215) and generate the security alert (block 220.) The nature

and type of the security alert generated, in accordance with the invention, can be in a variety of forms. Illustratively, the security alert generated by firewall 180 could be an immediate notification to the network administrator indicating that a particular client or clients within the network currently present a security risk. In a further embodiment of the invention, as probes are executed by various ones of the clients within the network, a log entry is made in a master file, e.g. stored in database 182, which can be accessed by the network administrator at regular intervals or a printed report could be generated from the log for review by the administrator.

Detailed Description Paragraph Right (10):
Advantageously, the invention provides a technique for determining whether particular clients with a computer network are universally configured in accordance with the desired network security features of the computer network. For example, one conventional security measure dictated by most network administrators is a policy that all users within a network, e.g., private network 130, disable certain features of their web browser software, e.g. Netscape.RTM., and in particular the Javascript interpreter feature of the web browser. Javascript is described in more detail, e.g., by D. Flanagan, Javascript The Definitive Guide, Second edition, O'Reilly & Associates, Sebastopol, Calif., 1997. Briefly, Javascript is a well-known interpreted programing language useful, e,g., in developing programs which relate to and involve web browsers and HTML. For example, when a web browser includes a Javascript interpreter, the browser enables executable content, e.g., programs, to be distributed over the Internet (and WWW) in the form of Javascript "scripts". When the script is loaded into a Javascript-enabled browser the script is executable and will produce particular output as defined by the Javascript instructions of the script. Thus, Javascript allows for the control over the web browser, and also the content of that which appears in a web page, e.g., HTML forms. As will be appreciated, these features which are enabled through the use of Javascript present serious network security risks.

Detailed Description Paragraph Right (11):
The import of the present invention in the web browser environment described above is detailed in the following illustrative embodiments. Turning our attention to FIGS. 1 and 3, private network 130 includes a plurality of users employing user terminals 165-1 through 165-4. As discussed previously, each user terminal can be configured with a web browser such as web browser 166 executing on user terminal 165-3. As will be readily understood, the configuration of user terminal 165-3 is easily replicated on each of the other user terminals within the private network but for purposes of clarity herein only one such configuration is shown in FIG. 1. Thus, in conformance with the security policy for private network 130, all web browsers are to have their Javascript interpreter disabled to prevent the execution of scripts which may be introduced from foreign sources, e.g., a public network, and subject the private network to various security risks. Of course, such a security measure is only effective if the users of the network comply. Typically, in most private networks there will exist, at any one time, particular user terminals which are not in compliance with the prescribed security measures. Thus, these non-complying user terminals represent a security risk to the entire network and a constant challenge to the network administrator for insuring full compliance with all security measures across the entire private network.

Detailed Description Paragraph Right (12):
As discussed previously, the invention provides a technique for determining whether particular clients with a computer network are universally configured in accordance with the desired network security features of the computer network. More particularly, firewall 180 is configured, as described above, in accordance with the invention to insert probes into the incoming communications traffic stream to private network 130. FIG. 3 shows an illustrative incoming communications traffic stream 300 and the insertion of an illustrative probe in accordance with the principles of the invention. In particular, communications traffic stream 300 includes a series of individual packets 300-1 through 300-n, e.g., TCP/IP packets, carrying data from public network 100 to private network 130. Thus, in accordance with the invention, firewall 180 monitors communication traffic stream 300 and randomly inserts probes into incoming files within particular ones of the packets. For example, packet 300-4 contains incoming file 305, illustratively a file having a series of HTML instructions 310. In accordance with the invention, virus prober 185 inserts probe 315, illustratively, at the end of HTML instructions 310. In accordance with various embodiments of the invention, probe 315 is inserted upon a first Internet access from a particular IP address (i.e., client) or browser type,

and thereafter virus probes are inserted at random intervals. Illustratively, probe 315 is a virus probe in trojan horse form, as previously discussed, wherein the insertion of probe 315 into file 305 results in edited file 325. Thereafter, edited file 325 proceeds in the transmission of communications traffic stream 300 to private network 130.

Detailed Description Paragraph Right (13):
Illustratively, probe 315 is a single Javascript instruction 320. As shown, Javascript instruction 320 is of the form " which, as discussed above, is an interpreted scripting language statement for controlling a web browser. Further, illustratively, "image1" is a unique string of characters for identifying probe 315. Basically, probe 315 is a trojan horse which directs the web browser to allocate an off-screen bitmap space, i.e., "new image( )" and download a small image, i.e., "image1". In accordance with various embodiments of the invention, the probes can either be stored in database 182 for access by virus prober 185 or stored locally within virus prober 185 itself In accordance with a further embodiment, probes can be downloaded by network administrators from a central source, e.g., the Internet, and added to the existing probe library. In accordance with the invention, if web browser 166 is in compliance with the illustrative network security feature which requires that all web browsers have their Javascript interpreter disabled, probe 315 will not execute and firewall 180 will not generate any security alert. However, in accordance with the invention if web browser 166 is misconfigured, probe 315 will execute causing web browser 166 to initiate a request for the image file, i.e., image1. As described previously, the mere request by web browser 166, in accordance with an embodiment of the invention, for a network resource is captured by firewall 180 thereby serving as the signal of a security alert. There is no reason for a properly configured web browser to ask for such a network resource, i.e., image1, unless it is improperly configured and outside of established network security measures. That is, execution of probe 315 means that web browser 166 is Javascript enabled which is not in compliance with the desired security measure of the private network 130 and therefore poses a security risk to the network.

Detailed Description Paragraph Right (14):
As described previously, a further embodiment of the invention employs a UDP packet as the signal back to the firewall when a security alert has occurred. In such an embodiment, file 305 is, illustratively, a file containing certain executable instructions. As is well-known, files having the extension ".exe" are binary executable files. Thus, in accordance with the invention, probe 315 will be inserted into file 305 at an appropriate location where it is known to be safe for overwriting a small number of bytes of file 305 for insertion of probe 315. In accordance with this embodiment of the invention, probe 315 will launch a UDP packet when a security alert occurs. Illustratively, the actual machine instructions inserted into file 305 are generated using, i.e. compiling, the following code segment written in the well-known C programming language:

Detailed Description Paragraph Left (1):
As will be appreciated by those skilled in the art, the above illustrative C program segment, after being compiled into machine code, is inserted as probe 315 into file 315 and will generate the desired UDP packet upon probe execution. That is, if probe 315 is executed on a particular user terminal, a UDP packet will be launched to firewall 180 as the signal indicating that the user terminal is a potential security risk.

CLAIMS:

1. A computer network security method, the method comprising the steps of:

monitoring a communications traffic stream of the computer network, the communications traffic stream including a plurality of files;

inserting a probe into at least one file of the plurality of files;

determining whether the probe is executed in the computer network; and

in response to the execution of the probe, identifying a location within the computer network where the execution of the probe occurred.

2. The method of claim 1 further comprising the step of:

generating a security alert containing at least the identified location within the computer network.

8. The method of claim 5 wherein the security alert is generated as a function of a UDP packet transmitted by the trojan horse.

9. A method for providing security in a private network, the private network having a plurality of user terminals, the method comprising the steps of:

monitoring a communications traffic stream between the private network and a public network, the communications traffic stream including a plurality of files, particular ones of the plurality of files destined for particular ones of the plurality of user terminals;

inserting at least one probe of a plurality of probes into the particular ones of the plurality of files;

determining whether the probe is executed by the particular one of the user terminals for which the file was destined; and

in response to the execution of the probe, identifying the particular one of the user terminals in which the execution of the probe occurred.

10. The method of claim 9 wherein the inserting the at least one probe step occurs in a firewall situated between the private network and the public network.

11. The method of claim 10 comprising the further step of:

transmitting a security alert from the probe to the firewall, the security alert containing an indication of at least the identified user terminal.

12. The method of claim 10 wherein the inserting the at least one probe step occurs as a function of a first access to the public network from at least one user terminal.

15. A method for use in a firewall which provides security between a private network and a public network, the method comprising the steps of:

monitoring a communications traffic stream transmitted between the private network and the public network, the communications traffic stream including a plurality of packets;

inserting a probe into at least one packet of the plurality of packets;

determining whether the probe is executed in the private network; and

in response to the execution of the probe, identifying a location within the private network where the execution of the probe occurred.

17. The method claim 16 wherein the identifying the location step further comprises transmitting a signal from the probe to the firewall indicating that the probe has executed.

18. The method claim 16 wherein the inserting the probe step occurs as a function of a first access to the public network from at least one user terminal.

19. A network security apparatus comprising:

a prober for inserting a plurality of probes into a plurality of packets exchanged between a private network and a public network; and

a processor for monitoring the plurality of packets and determining whether particular ones of the plurality of probes are executed in the private network.

20. The network security apparatus of claim 19 further comprising:

a database for storing the plurality of probes.

21. The network security apparatus of claim 19 further comprising a communications

channel for downloading the plurality of probes from a central source.

22. A network security method, the method comprising the steps of:

inserting a plurality of probes into an incoming communications stream of a private network; and

monitoring a plurality of user terminals in the private network for a execution of at least one probe of the plurality of probes.

24. The method of claim 22 wherein the monitoring the plurality of user terminals step further comprises transmitting a signal to a firewall indicating the execution of the at least one probe.

25. The method of claim 24 wherein the inserting the plurality of probes step occurs within a firewall.

27. The method of claim 26 wherein the inserting the plurality of probes step occurs as a function of a request from the private network for accessing a particular resource within the public network.

28. The method of claim 26 wherein the inserting the plurality of probes step occurs as a function of a first access to the public network from at least one user terminal.